

BAB II

LANDASAN TEORI

2.1 Perilaku Konsumen

Setiap konsumen merupakan pribadi individu yang berbeda-beda. Konsumen yang satu dengan yang lainnya mempunyai keinginan dan kebutuhan yang berbeda serta memiliki perilaku yang berbeda dalam memenuhi kebutuhan dan keinginan. Akan tetapi, dengan perilaku yang berbeda dalam kebutuhannya tersebut mereka masih memiliki kesamaan, yaitu sama-sama ingin memaksimalkan kepuasannya dalam mengonsumsi suatu barang atau jasa yang dibutuhkan.

Definisi perilaku menurut Schiffman dan Kantuk adalah sebuah kegiatan yang telah dilakukan oleh seseorang dalam mencari, membeli, menggunakan, mengevaluasi dan bertindak setelah mengonsumsi produk, jasa, ataupun ide yang diharapkan dapat memenuhi kebutuhannya.

Menurut Supranto, AMA (*American marketing Association*) mendefinisikan perilaku konsumen sebagai interaksi yang dinamis antara kognisi, afeksi, perilaku dan lingkungannya di mana seseorang melakukan kegiatan pertukaran dalam kehidupan mereka [5]. Dari sini dapat diambil 3 (tiga) hal penting, yaitu [5]:

- a. Perilaku konsumen bersifat dinamis, sehingga susah diprediksi.
- b. Melibatkan interaksi, seperti kognisi, afeksi, perilaku dan kejadian disekitar konsumen.
- c. Melibatkan pertukaran, seperti penukaran barang dan uang dari penjual kepada pembeli.

2.1.1 Faktor-faktor yang Mempengaruhi Pembelian Konsumen

Menurut Philipp Kotler, Gray Armstrong 2008 mengungkapkan terdapat 4 faktor yang dapat mempengaruhi pembelian konsumen dalam berbelanja, diantaranya adalah [6]:

2.1.1.1 Faktor Budaya

- a. Budaya, merupakan serangkaian nilai, persepsi, keinginan, dan perilaku dasar yang diterapkan oleh anggota masyarakat dari keluarga, lingkungan, dan instansi penting lainnya.
- b. Sub-budaya, merupakan kelompok manusia yang memiliki sistem nilai budaya yang sama berdasarkan pengalaman dan situasi kehidupan yang serupa.
- c. Kelas Sosial, merupakan pembagian kelompok masyarakat yang relative permanen dan relative teratur di mana anggota memiliki nilai, minat, dan perilaku yang serupa.

2.1.1.2 Faktor Sosial

- a. Kelompok, terdiri dari semua kelompok yang dapat mempengaruhi baik secara langsung maupun tidak langsung terhadap sikap dan perilaku seseorang tersebut.
- b. Keluarga.
- c. Peran dan Status. Peran yang terdiri dari sejumlah aktifitas yang diharapkan untuk dapat diterapkan atau dilakukan oleh orang-orang disekitarnya. Setiap peran membawa sebuah status yang menggambarkan suatu penghargaan umum terhadap peran tersebut dalam masyarakat.

2.1.1.3 Faktor Pribadi

- a. Umur dan Tata Siklus Hidup.
- b. Pekerjaan.
- c. Situasi ekonomi.
- d. Gaya Hidup, merupakan pola hidup seseorang yang tergambar dalam melakukan *activity*, *interest*, dan *opinion* (AIO) orang tersebut.
- e. Kepribadian dan Konsep Diri. Kepribadian dan psikologis yang membedakan seseorang dalam menghasilkan tanggapan secara konsisten dan terus-menerus terhadap lingkungan. Konsep diri adalah kepemilikan seseorang dapat menyumbang dan mencerminkan pada identitas diri mereka sendiri.

2.1.1.4 Faktor Psikologis

- a. Motivasi, adalah kebutuhan yang mendorong seseorang untuk mencari kepuasan atas kebutuhan tersebut.
- b. Persepsi, proses yang dilakukan untuk menyeleksi, mengatur dan menginterpretasikan informasi guna membentuk gambaran yang berarti.
- c. Pembelajaran, meliputi perubahan, perilaku dan pengalaman seseorang.
- d. Keyakinan dan Sikap. Keyakinan dalam berfikir deskriptif yang dipertahankan seseorang mengenai sesuatu. Sikap merupakan sebuah evaluasi, perasaan dan kecenderungan yang konsisten atas suka atau tidak seseorang terhadap suatu objek atau ide.

2.1.2 Variabel-Variabel dalam Mempelajari Perilaku Konsumen

Menurut Mangkunegara, terdapat 3 (tiga) variabel yang perlu diperhatikan dalam mempelajari perilaku konsumen, yaitu variabel stimulus, variabel respons, variabel Intervening [7].

2.1.2.1 Variabel Stimulus

Variabel stimulus merupakan variabel yang dimiliki di luar diri seseorang (faktor eksternal) yang sangat memberikan pengaruh dalam proses konsumsi.

Contoh: merek dan jenis barang, iklan, pramuniaga, penataan barang dan ruangan toko.

2.1.2.2 Variabel Respon

Variabel respons merupakan variabel yang menyatakan hasil aktivitas seseorang sebagai reaksi dari variabel stimulus, sehingga sangat bergantung pada faktor individu dan stimulus. Contoh: keputusan untuk membeli barang, memberikan penilaian terhadap barang, memberikan perubahan sikap terhadap suatu barang.

2.1.2.3 Variabel Intervening

Variabel intervening merupakan variabel yang berada diantara variabel stimulus dan variabel respons. Variabel ini merupakan faktor internal individu, termasuk faktor-faktor membeli, sikap terhadap suatu peristiwa, dan persepsi

terhadap suatu barang. Peranan variabel intervening adalah untuk memperbaiki respons.

2.2 *Data mining*

Secara singkat, *data mining* berarti menggali atau menemukan informasi dari sejumlah besar data. Dan secara luasnya, *data mining* adalah proses menemukan *interesting knowledge* dari sejumlah besar data yang tersimpan dalam database, *data warehouse*, atau media penyimpanan lainnya [8]. Sedangkan berdasarkan Gartner Group, *data mining* adalah sebuah proses untuk menemukan hubungan, pola, dan *tren* dengan memilah-milah sejumlah besar data yang tersimpan dalam media penyimpanan, menggunakan teknologi pengenalan pola serta statistika dan matematika [9].

Dengan berdasarkan pengertian diatas, maka dapat dikatakan jika *data mining* adalah sebuah proses mencari informasi dari data yang berjumlah besar dan tersimpan dalam media penyimpanan. *Data mining* bisa tercipta dikarenakan adanya keinginan untuk mencari pengetahuan atau informasi dari data yang tersimpan dalam jumlah yang banyak. Selain hal tersebut, dalam penerapannya terdapat beberapa faktor yang mendukung pengembangan *data mining*, yaitu:

1. Pertumbuhan Yang Besar Pada Data

Data mining, dalam penerapannya akan mengakses file yang besar. Hal ini dikarenakan hasil dari *data mining* akan menjadi dasar dalam pengambilan keputusan untuk melakukan tindakan tertentu, sehingga tingkat kebenaran informasi yang dihasilkan sangat diperlukan untuk mencapai tingkat kebenaran informasi yang tinggi, maka diperlukan data yang sangat besar untuk diolah. Seiring dengan perkembangan media penyimpanan, maka perkembangan jumlah data yang disimpan bergerak naik sangat cepat. Akan tetapi, jika terjadinya minim pengetahuan untuk mengolah data tersebut maka akan tercipta kondisi "*data rich but information poor*" [8], dimana tidak diketahuinya informasi dari data yang tersimpan tersebut.

2. Pertumbuhan Yang Luar Biasa Dalam Daya Komputasi Dan Kapasitas Penyimpanan Oleh Perangkat Keras

Penerapan *data mining* pada dasarnya akan membutuhkan kebutuhan sumber daya dan kemampuan komputasi yang sangat besar. Dengan perkembangan teknologi yang terjadi, mengakibatkan perkembangan yang sangat pesat bagi kemampuan komputasi dan kapasitas penyimpanan oleh perangkat keras. Sehingga dengan ketersediaan perangkat yang memiliki kemampuan tersebut, dapat menjadikan proses *data mining* menjadi cukup layak untuk dilakukan secara komersial.

3. Peningkatan Persaingan Untuk Meningkatkan Pasar Dalam Ekonomi Global

Seiring dengan persaingan bisnis yang semakin ketat, maka akan mendorong setiap perusahaan untuk selalu dapat berinovasi dan memberikan pelayanan-pelayanan atau promosi agar dapat bersaing dengan perusahaan lain. *Data mining* dalam peranannya dapat memberikan pengolahan informasi yang dapat menjadi pertimbangan bagi perusahaan dalam melakukan sebuah tindakan, seperti dalam memilih promosi, produk, dan pelayanan yang akan dipasarkan.

4. Peningkatan Terhadap Akses Ke Data Dengan Melalui Navigasi Web Atau Intranet

Perkembangan teknologi jaringan sampai sekarang, mengakibatkan kemudahan akses oleh data baik itu melalui jaringan internet atau intranet. Dengan kemudahan akses tersebut, maka mengakibatkan data transaksi mengenai akses data yang sangat besar. Sehingga dengan *data mining*, diharapkan dapat membantu dalam mengolah data transaksi tersebut agar dapat menjadi informasi yang berguna bagi perusahaan.

5. Pengembangan Terhadap Perangkat Lunak Untuk *Data mining*

Seiring dengan perkembangan dan kebutuhan akan *data mining*, mengakibatkan pertumbuhan perangkat lunak yang dapat mendukung *data mining* tersebut. Perangkat lunak yang dapat membantu dalam *data mining*, antara lain *Orange*, *RapidMiner*, *Weka*, *JhepWork*, *KNIME*, dan lain sebagainya. Dimana pada masing-masing perangkat lunak memiliki perbedaan masing-masing seperti bahasa pemrograman yang digunakan, bentuk tampilan, dan lain sebagainya.

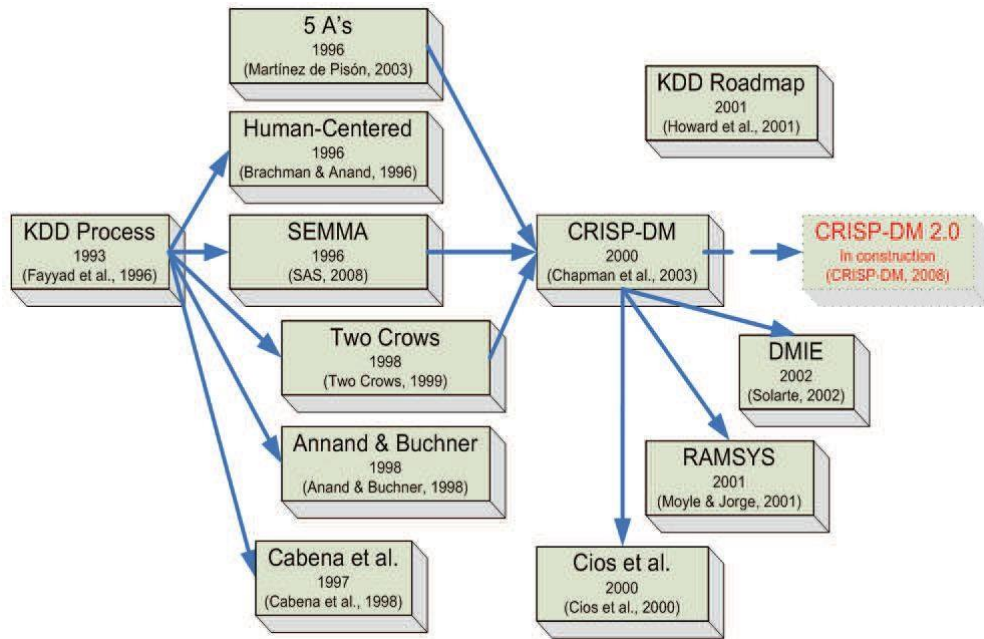
Dalam penerapannya, *data mining* dapat diterapkan dalam berbagai penyimpanan data [8].

Media penyimpanan yang disebut antara lain *relational database*, *data warehouses*, *transactional database*, *advanced database system*, *flat files*, *data streams*, dan *World Wide Web*. Dan dalam penerapannya pun, teknik dan algoritma yang digunakan untuk *data mining* pun berbeda satu sama lain antar setiap media penyimpana dan fungsi dari *data mining* tersebut. Algoritma *data mining* yang ada sekarang bermacam-macam, namun berdasarkan jurnal disebutkan 10 algoritma terbaik untuk *data mining*, antar lain C4.5, algoritma *k-means*, *support vector machine*, algoritma Apriori, algoritma *EM*, *PageRank*, *AdaBoost*, *kNN (k-nearest neighbor)*, *Naive Bayes*, *CART (Classification and Regression Tree)* [10].

Sedangkan untuk setiap algoritma menyesuaikan dengan fungsi dari *data mining*, yaitu fungsi karakterisasi, fungsi diskriminasi, fungsi asosiasi, fungsi prediksi, fungsi klasifikasi, dan fungsi cluster [8].

Pada pengembangannya hingga sekarang ini, terjadi beberapa perkembangan terhadap metodologi dalam penerapan *data mining*. Metodologi atau model proses merupakan suatu rangkaian tugas yang harus dilakukan untuk mengembangkan elemen tertentu, serta unsur-unsur yang diproduksi dalam setiap tugas(output) dan dan unsur-unsur yang diperlukan untuk melakukan tugas(input). Dengan tujuan untuk membuat proses agar dapat berulang, dikendalikan, dan terukur [11].

Perkembangan metodologi pada *data mining* terjadi mulai pada metodologi KDD (*Knowledge Discovery from Data*) pada awal tahun 1990-an. Setelah itu terjadi perkembangan metodologi yang menyusul setelahnya. Berikut merupakan gambaran perkembangan metodologi *data mining*.



Gambar 2. 1 Perkembangan Metodologi *Data mining* [11]

Pada gambaran diatas, diketahui perkembangan metodologi dimulai pada metodologi KDD. Selanjutnya dengan berdasar KDD dikembangkan beberapa metodologi dengan melakukan beberapa perubahan namun secara garis besar memiliki fitur serupa, yaitu *Human-centered*, SEMMA, *Two Crows*, *Anand & Buchner* dan *Cabena*.

Di lain sisi, metodologi 5 A's merupakan metodologi yang menjadi pelopor dari CRISP-DM, dimana siklus yang dikerjakan memiliki kemiripan dengan siklus pada CRISP-DM. Selanjutnya CRISP-DM menjadi dasar pengembangan CIOS yang lebih fokus pada kebutuhan riset akademik, RAMSYS (*Rapid collaborative data mining SYStem*) dan DMIE (*Data mining for Industrial Engineering*).

Metodologi tersebut tetap didasarkan pada CRISP-DM dan mempertahankan fase yang sama secara generik. Pada tabel berikut ini akan dijelaskan mengenai tahapan pada beberapa metodologi.

Tabel 2. 1 Perbandingan Metodologi [11]

Model	Fayyad et al.	Cabena et al.	Anand & Buchner	CRISP-DM	Cios et al.	
No of steps	9	5	8	6	6	
Steps	Developing and Understanding of the Application Domain	Business Objectives Determination	Human Resource Identification Problem Specification	Business Understanding	Understanding the Data	
	Creating a Target Data Set	Data Preparation	Data Prospecting	Data Understanding	Understanding the Data	
	Data Cleaning and Pre-processing		Domain Knowledge Elicitation			
	Data Reduction and Projection		Methodology Identification	Data Preparation	Preparation of the data	
	Choosing the DM Task	Data Pre-processing				
	Choosing the DM Algorithm	DM	DM	Pattern Discovery	Modeling	DM
	Interpreting Mined Patterns	Domain Knowledge Elicitation	Knowledge Post-processing	Evaluation	Evaluation of the Discovered Knowledge	
	Consolidating Discovered Knowledge	Assimilation of Knowledge		Deployment	Using the Discovered Knowledge	

Walau jumlah tahapan dan tahapan yang dilakukan berbeda-beda, namun secara garis besar tahapan pada setiap metodologi *data mining* adalah sebagai berikut :

1. *Pre-processing*

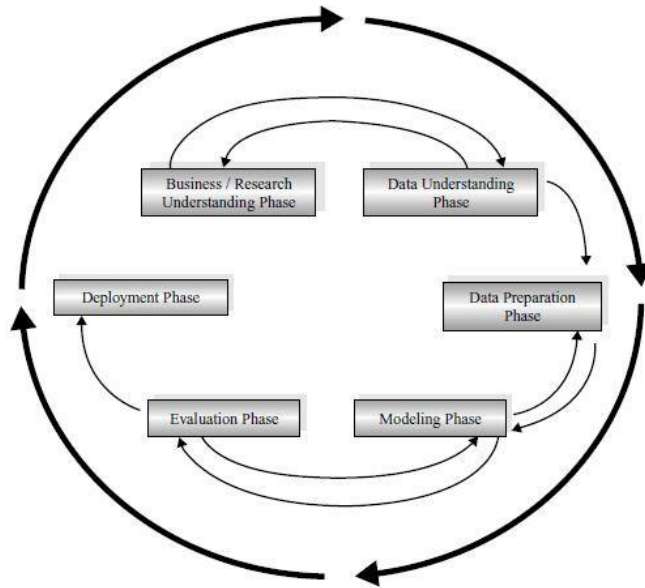
Pada tahapan ini dilakukan persiapan mengenai *data mining* yang akan dilakukan. Seperti pengenalan bisnis, pengenalan data, dan persiapan data.

2. *Data mining*

Pada tahap ini dilakukan kegiatan *data mining* dengan berdasarkan metode dan algoritma yang ditentukan.

3. *Post-processing*

Tahapan ini adalah tahapan setelah melakukan proses *data mining*.



Gambar 2. 2 Tahapan Metodologi CRISP-DM[9]

CRISP-DM merupakan singkatan dari *Cross Industry Standard Process for Data mining*. CRISP-DM merupakan standarisasi *data mining* yang disusun oleh tiga pengagagas *data mining* market. Yaitu Daimler Chrysler (Daimler-Benz), SPSS (ISL), NCR. Pada metodologi ini dilakukan pembagian siklus untuk proses *data mining* menjadi 6 tahap, diantaranya [9] :

1. *Business Understanding*

Pada tahap ini berfokus pada pemahaman mengenai tujuan dari proyek dan kebutuhan secara persepektif bisnis, kemudian mengubah hal tersebut menjadi sebuah permasalahan *data mining* dan rencana awal untuk mencapai tujuan tersebut.

Kegiatan yang dilakukan antara lain: menentukan tujuan dan persyaratan dengan jelas secara keseluruhan, menerjemahkan tujuan tersebut serta menentukan pembatasan dalam perumusan masalah *data mining*, dan selanjutnya mempersiapkan strategi awal untuk mencapai tujuan tersebut.

2. *Data Understanding*

Pada tahap ini dilakukan pengumpulan terhadap data, lalu kemudian mempelajari data tersebut dengan tujuan untuk mengenal data, melakukan identifikasi dan mengetahui kualitas dari data, serta mendeteksi subset yang

menarik dari data yang dapat dijadikan hipotesa bagi informasi yang tersembunyi.

3. *Data Preparation*

Pada tahap ini dilakukan persiapan mengenai data yang akan digunakan pada tahap berikutnya. Kegiatan yang dilakukan antara lain: memilih kasus dan parameter yang akan dianalisis (*Select Data*), melakukan transformasi terhadap parameter tertentu (*Transformation*), dan melakukan pembersihan data agar data siap untuk tahap *modeling* (*Cleaning*).

4. *Modeling*

Pada tahap ini dilakukan penentuan terhadap teknik *data mining*, alat bantu *data mining*, dan algoritma *data mining* yang akan diterapkan. Lalu selanjutnya adalah melakukan penerapan teknik dan algoritma *data mining* tersebut kepada data dengan bantuan alat bantu. Jika diperlukan penyesuaian data terhadap teknik *data mining* tertentu, dapat kembali ke tahap persiapan data.

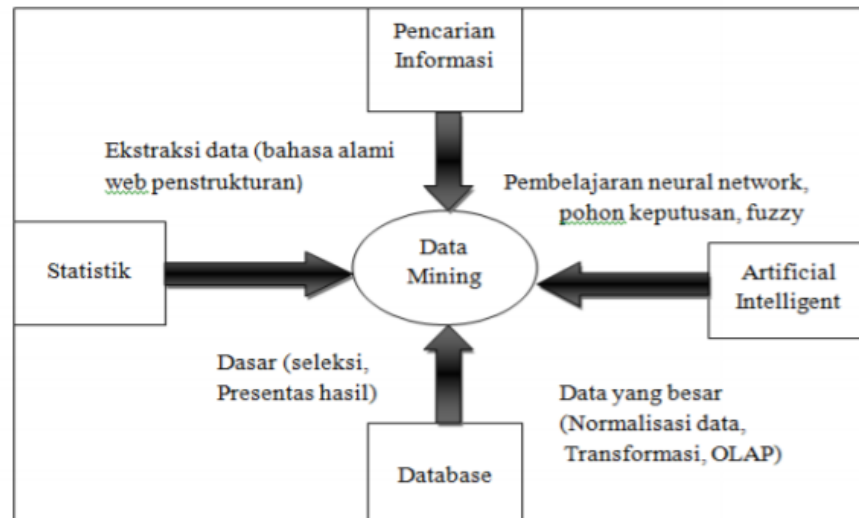
5. *Evaluation*

Melakukan interpretasi terhadap hasil dari *data mining* yang dihasilkan dalam proses pemodelan pada tahap sebelumnya. Evaluasi dilakukan terhadap model yang diterapkan pada tahap sebelumnya dengan tujuan agar model yang ditentukan dapat sesuai dengan tujuan yang ingin dicapai dalam tahap pertama.

6. *Deployment*

Melakukan penyusunan laporan terhadap hasil yang didapat dari evaluasi pada tahap sebelumnya atau dari proses *data mining* yang dilakukan secara keseluruhan.

Tujuan *data mining* adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat. *Data mining* bukanlah suatu bidang yang baru. Salah satu kesulitan untuk mendefinisikan *data mining* adalah kenyataan bahwa *data mining* mewarisi banyak aspek dan teknik dari bidang ilmu yang sudah mapan terlebih dahulu. Disana ditunjukkan bahwa *data mining* memiliki akar yang panjang dari bidang ilmu seperti Kecerdasan Buatan (*Artificial Intellegent*), *Machine Learning*, *Statistic*, *Database*, dan juga *Information Retrieval* [12].



Gambar 2. 3 Fungsi *Data mining* [9]

Selanjutnya, terdapat beberapa fungsi atau tugas dari *data mining* yang biasanya dikerjakan dalam proyek *data mining*. Fungsi tersebut antara lain [9] :

1. *Description*

Data mining dapat berfungsi untuk melakukan deskripsi pola dan trend pada kumpulan data, dengan melakukan deskripsi akan didapat penjelasan yang mungkin untuk tren atau pola tersebut. *Data mining* harus bersifat se-transparan mungkin, karena hasil dari *data mining* harus menjelaskan pola yang jelas.

2. *Estimation*

Fungsi *estimation* memiliki kemiripan dengan fungsi *classification*, yaitu bertujuan untuk menentukan nilai dari atribut output yang belum diketahui. Namun berbeda dengan *classification*, atribut output pada *estimation* lebih bersifat numerik daripada kategori.

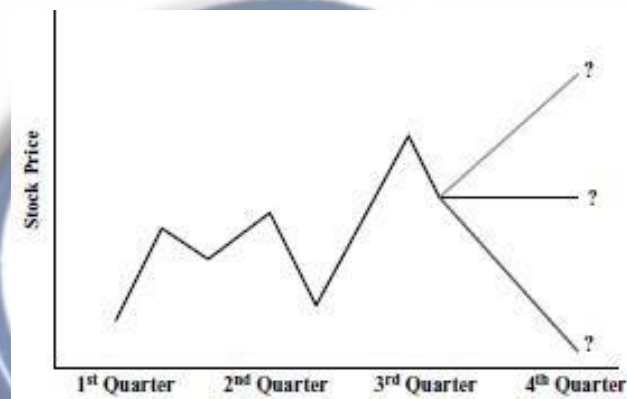
Contoh kasus yang menerapkan *estimation* antara lain:

- a. Memperkirakan rata-rata nilai (IPK) dari seorang mahasiswa pasca sarjana dengan didasarkan pada IPK ketika sarjana.
- b. Memperkirakan jumlah uang yang dikeluarkan untuk berbelanja keperluan sekolah ketika tahun ajaran baru dimulai oleh sebuah keluarga.

6. Prediction

Tujuan dari fungsi ini adalah untuk menemukan kemungkinan hasil di masa depan dengan berdasarkan tindakan sekarang. Dimana atribut output yang dihasilkan bisa berupa kategori atau numerik. Berikut merupakan contoh kasus yang menerapkan fungsi *prediction* [8]:

- Memprediksi pemenang dari kejuaraan dengan berdasarkan perbandingan statistik tim.
- Memprediksi persentase kenaikan tingkat kecelakaan lalu lintas pada tahun depan jika batasan kecepatan meningkat.
- Memprediksi harga saham 3 bulan ke depan.



Gambar 2. 4 Prediksi Harga Saham 3 Bulan[8]

7. Classification

Klasifikasi merupakan sebuah proses untuk mencari model atau fungsi yang menjelaskan dan membedakan kelas atau konsep dari data, dengan tujuan untuk menggunakan model dan melakukan prediksi dari kelas suatu objek dimana tidak diketahui label dari kelas tersebut.

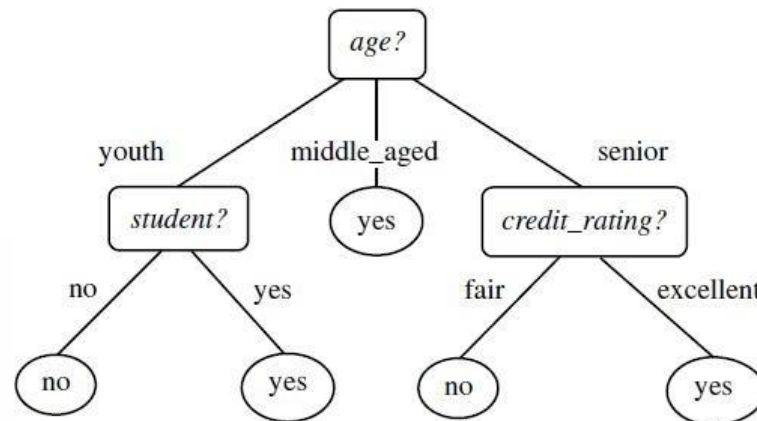
Berikut merupakan contoh dari penerapan fungsi klasifikasi:

- Menentukan apakah transaksi dengan kartu kredit tersebut adalah penipuan.
- Menilai apakah pengajuan kredit yang diajukan memiliki resiko kredit yang baik atau buruk.
- Mendiagnosis apakah terdapat penyakit tertentu.

Dalam melakukan representasi terhadap klasifikasi, terdapat beberapa model yang dapat digunakan. Representasi tersebut bisa dengan

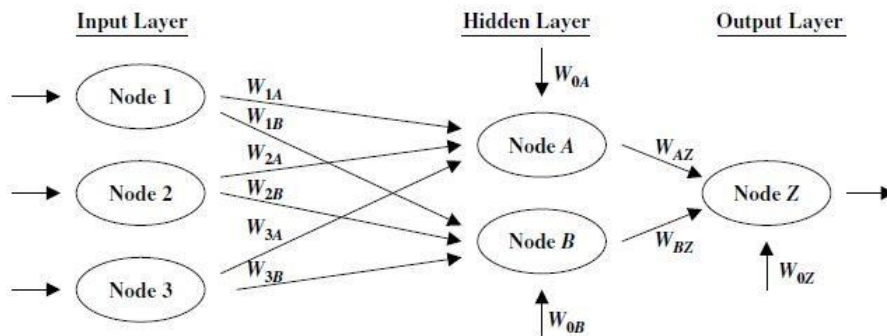
menggunakan *classification (IF-THEN) rules*, *decision trees*, *mathematical formulae*, atau *neural networks*.

Decision tree merupakan sebuah diagram alur yang berbentuk seperti pohon, dimana setiap cabang merepresentasikan output dari pengujian dan daun pohon merepresentasikan kelas atau distribusi dari kelas.



Gambar 2. 5 Contoh Decision Tree [8]

Neural network berbentuk seperti sebuah koleksi unit proses yang berbentuk neuron dengan *weighted connections* diantara setiap unit.



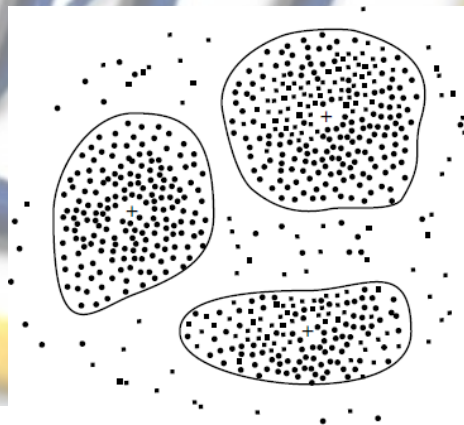
Gambar 2. 6 Contoh Neural Network[8]

Dalam pemanfaatannya terdapat banyak metode yang dapat digunakan untuk melakukan model klasifikasi, seperti *support vector machine*, *naive Bayesian classification*, dan *k- nearest neighbor classification*.

8. *Clustering*

Clustering berkaitan dengan pengelompokan dari records, penelitian, atau kasus ke dalam kelas yang memiliki kesamaan objek. *Clustering* berbeda dengan *classification*, dikarenakan pada *clustering* tidak ada target parameter yang digunakan untuk *clustering*. Algoritma pada *clustering* akan melakukan segmentasi data pada seluruh data menjadi sub-kelompok yang relatif memiliki kesamaan. Berikut merupakan contoh dari penerapan fungsi *clustering*:

- a. Menentukan target pemasaran produk untuk bisnis kecil yang tidak memiliki anggaran pemasaran yang besar.
- b. Sebagai alat pengurangan dimensi data ketika set data memiliki ratusan atribut.
- c. Mengelompokkan pelanggan dengan berdasarkan karakterisasi pola pembeliannya, sehingga dapat membantu pemasaran untuk strategi pemasaran.



Gambar 2. 7 Contoh Penerapan *Clustering* lokasi pelanggan di sebuah kota [8]

Dalam penerapannya, banyak algoritma yang dapat digunakan untuk melakukan clustering, seperti *k-means algorithm*, dan *self-organizing maps (SOM)*.

9. *Association*

Asosiasi dalam penerapan *data mining* memiliki tugas untuk menemukan atribut yang memiliki keterkaitan atau mencari hubungan atau suatu item. Sebagai contohnya dalam himpunan transaksi, dimana dari himpunan

transaksi tersebut mungkin akan ditemukan suatu hubungan seperti berikut, pada sebuah supermarket ditemukan bahwa terdapat 1000 pelanggan yang melakukan transaksi pembelian dimana 200 pelanggan membeli popok, dan dari 200 pelanggan tersebut diketahui 50 pelanggannya membeli bir. Dalam bahasa asosiasi hal tersebut dibaca menjadi “Jika membeli popok, maka membeli bir” dengan *support* $200/1000 = 20\%$ dan *confidence* $50/200 = 25\%$. *Support* dalam hal ini berarti persentase jumlah transaksi dalam analisis yang menggambarkan bahwa kedua produk telah dipesan dalam satu transaksi yang sama. Sedangkan *confidence* berarti persentase jumlah pelanggan yang akan membeli produk ke-2 jika membeli produk pertama, dalam hal ini berarti jika pembeli membeli popok, maka ada peluang 25% pembeli tersebut juga akan membeli bir.

Dengan demikian, untuk dapat menemukan hubungan antar item tersebut mungkin diperlukan pembacaan terhadap data transaksi secara berulang terhadap data transaksi tersebut. Dikarenakan data transaksi tersebut berjumlah sangat besar maka diperlukan algoritma yang efisien untuk dapat mengerjakannya.

Selain itu terdapat beberapa contoh penerapan fungsi asosiasi, antara lain:

- a. Menginvestigasi proporsi pelanggan yang akan merespon positif terhadap tawaran dari perusahaan ponsel.
- b. Memprediksi degradasi dari jaringan telekomunikasi.
- c. Mendapatkan keterkaitan antar parameter yang berpengaruh pada peluang berhasilnya lulus ujian dari seorang pelajar ketika menghadapi ujian akhir.

Untuk menerapkan asosiasi terdapat beberapa algoritma yang dapat digunakan, yaitu *Apriori algorithm*, dan *GRI (generalized rule induction) algorithm*.

2.3 Association Rule

Association Rule sering juga disebut dengan *Market basket analysis* karena kegunaannya dalam pembuatan model perilaku pembelian oleh konsumen, dan menganalisis perilaku konsumen. *Association Rule* adalah suatu prosedur dalam analisis keranjang pasar (*Market basket analysis*) untuk menemukan aturan asosiasi yang memenuhi syarat minimum untuk *support* dan *confidence* (*minimum support* dan *minimum confidence*) [13]. Analisis asosiasi atau *association rule mining* adalah teknik *data mining* untuk menemukan aturan asosiatif antara suatu kombinasi *item*. Algoritma aturan asosiasi akan menggunakan data latihan sesuai dengan pengertian *data mining* untuk menghasilkan pengetahuan.

Penting tidaknya suatu aturan asosiasi dapat diketahui dengan adanya dua parameter, yaitu *support* dan *confidence*. *Support* (nilai penunjang) adalah presentase kombinasi *Itemset* produk dalam *database*. Sedangkan *confidence* (nilai kepastian) adalah nilai untuk menentukan kuatnya hubungan antar *item* dalam aturan asosiasi. Dalam asosiasi juga terdapat istilah *antecedent* dan *consequent*. *Antecedent* untuk mewakili pernyataan “jika” dan *consequent* untuk mewakili pernyataan “maka”. Dalam Analisa ini, *antecedent* dan *consequent* adalah sekelompok *item* yang tidak mempunyai hubungan secara langsung.

Association Rule merupakan salah satu metode yang bertujuan mencari pola yang sering muncul di antara banyak transaksi, dimana setiap transaksi terdiri dari beberapa *item* sehingga metode ini akan mendukung sistem rekomendasi melalui penemuan pola antar *item* dalam transaksi-transaksi yang terjadi metodologi dasar analisis asosiasi terbagi menjadi dua tahap [14] :

2.3.1 Analisa Pola Frekuensi Tinggi

Tahap ini mencari kombinasi *item* yang memenuhi syarat minimum dari nilai *support* dalam *database*. Nilai *support* sebuah *Itemset* di peroleh dengan rumus berikut:

$$Support(A) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ transaksi} \quad (1)$$

Sedangkan nilai *support* dari 2 *Itemset* di peroleh rumus berikut ;

$$\text{Support } (A, B) = P(A \cap B)$$

$$\text{Support } (A \cap B) = \frac{\sum \text{Jumlah transaksi mengandung } A \text{ dan } B}{\sum \text{Total transaksi}} \quad (2)$$

2.3.2 Pembentukan Aturan Asosiatif

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiasi yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiatif $A \rightarrow B$. Nilai *confidence* dari aturan $A \rightarrow B$ diperoleh dari rumus berikut:

$$\text{Confidence} = P(A|B) = \frac{\sum \text{Jumlah transaksi mengandung } A \text{ dan } B}{\sum \text{Total transaksi mengandung } A} \quad (3)$$

2.4 Korelasi

Korelasi adalah istilah statistik yang menyatakan derajat hubungan linear antara dua variabel atau lebih, yang ditemukan oleh Karl Pearson pada awal 1900. Oleh sebab itu terkenal dengan sebutan Korelasi Pearson Product Moment (PPM)[15].

Korelasi Pearson Product Moment (PPM) sering disingkat Korelasi merupakan salah satu teknik analisis statistik yang paling banyak digunakan oleh para peneliti. Karena peneliti umumnya tertarik terhadap peristiwa-peristiwa yang terjadi dan mencoba untuk menghubungkannya. Besarnya angka korelasi disebut koefisien korelasi dinyatakan dengan lambang r .

Hubungan antara dua variabel di dalam teknik korelasi bukanlah dalam arti hubungan sebab akibat (timbal balik), melainkan hanya merupakan hubungan searah saja. Akibatnya, dalam korelasi dikenal penyebab dan akibatnya. Data penyebab atau yang mempengaruhi disebut variabel bebas (*independent*) dan data akibat atau yang dipengaruhi disebut variabel terikat (*dependent*). Variabel bebas (*independent*) dilambangkan dengan huruf X atau $X_1, X_2, X_3 \dots X_n$ (tergantung banyaknya variabel bebas). Variabel terikat (*dependent*) dilambangkan dengan huruf Y.

2.4.1 Kegunaan Korelasi Pearson Product Moment (PPM)

Adapun kegunaan dari korelasi ini adalah sebagai berikut [15]:

1. Untuk menyatakan ada atau tidaknya hubungan yang signifikan antara variabel satu dengan yang lainnya.
2. Untuk menyatakan besarnya sumbangan variabel satu terhadap yang lainnya yang dinyatakan dalam persen. Dengan demikian, maka r^2 disebut koefisien determinasi atau koefisien penentu. Hal ini disebabkan $r^2 \times 100\%$ terjadi dalam variabel terikat Y yang mana ditentukan oleh variabel X.

2.4.2 Pola atau Bentuk Hubungan antara 2 Variabel

Korelasi yang terjadi antara dua variabel adalah sebagai berikut [15]:

1. Korelasi Linear Positif (+1)

Perubahan salah satu nilai variabel diikuti perubahan nilai variabel yang lainnya secara teratur dengan arah yang sama. Jika nilai variabel X mengalami kenaikan, maka variabel Y akan ikut naik. Jika nilai variabel X mengalami penurunan, maka variabel Y akan ikut turun.

Apabila nilai koefisien korelasi mendekati +1 (positif satu) berarti pasangan data variabel X dan variabel Y memiliki korelasi linear positif yang kuat/erat/sempurna.

2. Korelasi Linear Negatif (-1)

Perubahan salah satu nilai variabel diikuti perubahan nilai variabel yang lainnya secara teratur dengan arah yang berlawanan. Jika nilai variabel X mengalami kenaikan, maka variabel Y akan turun. Jika Nilai variabel X mengalami penurunan, maka nilai variabel Y akan naik.

Apabila nilai koefisien korelasi mendekati -1 (negatif Satu) maka hal ini menunjukkan pasangan data variabel X dan variabel Y memiliki korelasi linear negatif yang kuat/erat/sempurna.

3. Tidak Berkorelasi (0)

Kenaikan nilai variabel yang satunya *bisa diikuti* dengan penurunan variabel lainnya atau *kadang-kadang diikuti* dengan kenaikan variabel yang lainnya. Arah hubungannya tidak teratur, bisa searah atau pun berlawanan.

Apabila Nilai Koefisien Korelasi mendekati 0 (Nol) berarti pasangan data variabel X dan variabel Y memiliki korelasi yang sangat lemah atau berkemungkinan tidak berkorelasi.

2.4.3 Menghitung Nilai Koefisien Korelasi (r)

Untuk menghitung korelasi antara variabel bebas (*independent*) dengan variabel terikat (*dependent*) dapat digunakan rumus sebagai berikut [15]:

$$r_{yx} = \frac{n \sum X_i Y_i - (\sum X_i)(\sum Y_i)}{\sqrt{(n \sum X_i^2 - \sum X_i^2)(n \sum Y_i^2 - \sum Y_i^2)}}$$

Dengan :

r_{yx} = Koefisien korelasi antara Y dan X

X_i = Variabel bebas (*independent*)

Y = Variable terikat (*dependent*)

Nilai r selalu terletak antara -1 dan 1, sehingga nilai r tersebut dapat ditulis : $-1 \leq r \leq +1$. Untuk $r = +1$, berarti ada korelasi positif sempurna antara variabel X dan variabel Y sebaliknya jika $r = -1$, berarti korelasi negatif sempurna antara variabel X dan variabel Y, sedangkan $r = 0$, berarti tidak ada korelasi antara X dan Y.

Jika kenaikan didalam suatu variabel diikuti dengan kenaikan di dalam variabel lain, maka dapat dikatakan bahwa kedua variabel tersebut mempunyai korelasi yang positif. Tetapi jika kenaikan di dalam suatu variabel diikuti oleh penurunan di dalam variabel lain, maka dapat dikatakan bahwa variable tersebut mempunyai korelasi yang negatif. Dan jika tidak ada perubahan pada variabel walaupun variabel lainnya berubah maka dikatakan bahwa kedua variabel tersebut tidak mempunyai hubungan. Interpretasi harga r akan disajikan dalam **tabel 2.2** berikut [15]cocoh:

Tabel 2. 2 Interpretasi Koefisien Korelasi (r)

R	Interpretasi
0	Tidak berkorelasi
0,01 – 0,20	Sangat rendah
0,21 – 0,40	Rendah
0,41 – 0,60	Agak rendah
0,61 – 0,80	Cukup
0,81 – 0,99	Tinggi
1	Sangat tinggi

2.3.4 Uji Keberartian Koefisien Korelasi

Langkah-langkah untuk uji keberartian koefisien korelasi adalah sebagai berikut [15]:

1. Menentukan Hipotesa

H_0 : Tidak terdapat hubungan yang signifikan antara variabel bebas (*independent*) dengan variabel terikat (*dependent*).

H_1 : Terdapat hubungan yang signifikan antara variabel bebas (*independent*) dengan variabel terikat (*dependent*).

2. Menentukan Taraf Nyata (Significant Level)

$$\alpha = 0,05$$

$$dk = n-2$$

3. Menentukan Kriteria Pengujian

H_0 ditolak jika $t_{hitung} > t_{tabel}$ atau $t_{hitung} < -t_{tabel}$

H_0 diterima jika $-t_{tabel} \leq t_{hitung} \leq t_{tabel}$

4. Menentukan Nilai Uji Statistik t_{hitung} dapat dicari dengan rumus:

$$t_{hitung} = r \sqrt{\frac{n-2}{1-r^2}}$$

5. Membuat Kesimpulan

Menyimpulkan tentang penerimaan atau penolakan H_0 (sesuai dengan kriteria pengujiannya).

2.5 Algoritma Apriori

Pada jurnal “*Top 10 algorithms in data mining*” disebutkan bahwa algoritma Apriori merupakan algoritma yang paling banyak digunakan dalam *data mining* dan menempati peringkat ke-4 (empat) [10]:

1. C4.5
2. K-means
3. SVM (*Support Vector machines*)
4. Algoritma Apriori
5. EM (*Expectation Maximazation*)
6. Algoritma PageRank
7. Algoritma AdaBoost
8. K-Nearst Neighbor
9. Naive Bayes
10. *Classification and Regression Trees*

Algoritma Apriori adalah algoritma analisis keranjang pasar yang digunakan untuk menghasilkan aturan asosiasi, dengan pola “*if-then*”. Algoritma Apriori menggunakan pendekatan iteratif yang dikenal dengan *level-wise search*, dimana k-kelompok produk digunakan untuk mengeksplorasi (k+1)-kelompok produk atau (k+1)-*Itemset* [16]. Beberapa istilah yang digunakan dalam algoritma Apriori antara lain [16] :

- a. *Support* (dukungan): *probabilitas* pelanggan membeli beberapa produk secara bersamaan dari seluruh transaksi. *Support* untuk aturan “ $X \Rightarrow Y$ ” adalah *probabilitas* atribut atau kumpulan atribut X dan Y yang terjadi bersamaan.
- b. *Confidence* (tingkat kepercayaan) yaitu *probabilitas* kejadian beberapa produk dibeli bersamaan dimana salah satu produk sudah pasti dibeli. Contoh: jika ada n transaksi dimana X dibeli, dan ada m transaksi dimana X dan Y dibeli bersamaan, maka *confidence* dari aturan if X then Y adalah m/n .

- c. *Minimum support* yaitu parameter yang digunakan sebagai batasan frekuensi kejadian atau *support count* yang harus dipenuhi suatu kelompok data untuk dapat dijadikan aturan.
- d. *Minimum confidence* yaitu parameter yang mendefinisikan minimum level dari *confidence* yang harus dipenuhi oleh aturan yang berkualitas.
- e. *Itemset* yaitu kelompok produk.
- f. *Support count* yaitu frekuensi kejadian untuk sebuah kelompok produk atau *Itemset* dari seluruh transaksi.
- g. Kandidat *Itemset* yaitu *Itemset-Itemset* yang akan dihitung *support count*-nya.
- h. *Large Itemset* yaitu *Itemset* yang sering terjadi, atau *Itemset-Itemset* yang sudah melewati batas *minimum support* yang telah diberikan.

2.4.1 Mencari Akurasi pada Algoritma Apriori

Mencari akurasi pada metode apriori itu berdasarkan nilai confidence dan conviction. Nilai Confidence adalah nilai akurasi atau biasa disebut nilai kepastian, rules tersebut pasti (akurat) atau tidak, tergantung dari besar kecilnya confidence. Jika nilai Confidence kecil, maka dapat di pastikan rules tersebut kurang akurat atau kurang pasti, tetapi jika nilai Confidence besar maka dapat dipastikan rules tersebut akurat atau pasti. Nilai confidence menentukan kuatnya hubungan antar item dalam apriori.

Jika nilai confidence sudah ditentukan, maka selanjutnya menentukan nilai conviction. Nilai conviction dapat digunakan untuk mengetahui tingkat akurasi dan kekuatan rule yang terbentuk. Nilai range pada Conviction ini, berada pada nilai $0,5, \dots, 1, \dots, \infty$ dengan ketentuan conviction dianggap memiliki nilai tak terhingga (infinite) apabila nilai dari confidence ($A \rightarrow B$) sama dengan 1. Conviction menghasilkan nilai rule yang semakin menjauh dari 1 bahkan sampai tak hingga (∞), maka akan di anggap semakin akurat.

Conviction merupakan suatu nilai yang menentukan akurasi dari *rules* yang telah dibentuk. Tingkat keakuratan dari sebuah *rules*, itu ditentukan berdasarkan besar kecilnya nilai *conviction*. Semakin besar nilai *conviction* maka akan semakin akurat pula *rules* yang dihasilkan. Beberapa pengukuran untuk mengetahui tingkat akurasi dan kekuatan *rules* yang terbentuk dengan mengetahui nilai *conviction*. Rumus dari *Conviction* adalah sebagai berikut :

$$\text{Conviction}(A \rightarrow B) = \frac{1 - \text{Support}(B)}{1 - \text{Confidence}(A \rightarrow B)}$$

Diketahui :

A,B : Item

Support(B) : Nilai *Support* dari Item B

Confidence : Nilai *Confidence* aturan asosiatif (A→B)

Contoh algoritma Apriori :

Terdapat data transaksi seperti yang di tunjukan pada tabel 2.1. Sebagai perumpamaan yang diinginkan *minimum support* : 50 % (2 dari 4 transaksi).

Tabel 2. 3 Data Item Transaksi

Transaksi ID	Itemset
891600	<i>Liquid Foundation, Face Powder, Cleansing Foam</i>
891606	<i>Eyelineer Blue, Face Powder, Eye Makeup Remover</i>
891666	<i>Liquid Foundation, Eyelineer Blue, Face Powder, Eye Makeup, Remover</i>
991606	<i>Eyelineer Blue, Eye Makeup Remover</i>

Berikut langkah-langkahnya :

1. Mencari nilai *support* untuk masing-masing *Itemset* L1 (large 1-*Itemset*).

Tabel 2. 4 Nilai *support* (1-*Itemset*)

Itemset	Support
<i>Liquid Foundation</i>	50%
<i>Eyelineer Blue</i>	75%

<i>Face Powder</i>	75%
<i>Cleansing Foam</i>	25%
<i>Eye Makeup Remover</i>	75%

2. Mencari kandidat *Itemset* untuk L2.
 - a. Menggabungkan *Itemset* pada L1 (algoritma Apriori-gen). lalu hapus yang tidak ada dalam *Itemset* di lihat dari tabel 2.3. *Itemset* (*Eyliner Blue*, *Cleansing Foam*, *Cleansing Foam*, *Eye Makeup Remover*) dihapus karena tidak ada dalam *Itemset*.
3. Hitung nilai *support* untuk masing-masing *Itemset*. Hasilnya dapat ditunjukkan pada table 2.3.

Tabel 2. 5 Nilai *Support* (2-*Itemset*)

<i>Itemset</i>	<i>Support</i>
<i>Liquid Foundation, Eyeliner Blue</i>	25%
<i>Liquid Foundation, Face Powder</i>	50%
<i>Liquid Foundation, Cleansing Foam</i>	25%
<i>Liquid Foundation, Eye Makeup Remover</i>	25%
<i>Eyeliner Blue, Face Powder</i>	50%
<i>Eyeliner Blue, Eye Makeup Remover</i>	75%
<i>Face Powder, Cleansing Foam</i>	25%
<i>Face Powder, Eye Makeup Remover</i>	50%

4. Tentukan *Itemset* yang memenuhi *minimum support* L2 (large 2-*Itemset*).

Tabel 2. 6 2-*Itemset* Memenuhi Min. *Support*

<i>Item set</i>	<i>support</i>
<i>Liquid Foundation, Face Powder</i>	50%
<i>Eyeliner Blue, Face Powder</i>	50%

<i>Eyeliners Blue, Eye Makeup Remover</i>	75%
<i>Face Powder, Eye Makeup Remover</i>	50%



5. Ulangi Langkah 2 sampai 4. Gabungkan pada L2 & L2.

Tabel 2. 7 Anggota 3-Itemset

<i>Itemset</i>	Hasil Gabungan (3 Itemset)
<i>Liquid Foundation, Face Powder + Eyeliner Blue, Face Powder</i>	A C B
<i>Liquid Foundation, Face Powder + Eyeliner Blue, Eye Makeup Remover</i>	A C B, A C E, A B E
<i>Liquid Foundation, Face Powder + Face Powder, Eye Makeup Remover</i>	A C E
<i>Eyeliner Blue, Face Powder + Eyeliner Blue, Eye Makeup Remover</i>	B C E
<i>Eyeliner Blue, Face Powder + Face Powder, Eye Makeup Remover</i>	B C E
<i>Eyeliner Blue, Eye Makeup Remover + Face Powder, Eye Makeup Remover</i>	B C E

6. Hitung *support* dari setiap kandidat *Itemset* L3.

Tabel 2. 8 Nilai *Support* untuk 3-Itemset

<i>Itemset</i>	<i>Support</i>
<i>Liquid Foundation, Eyeliner Blue, Face Powder</i>	25%
<i>Liquid Foundation, Eyeliner Blue, Eye Makeup Remover</i>	25%
<i>Eyeliner Blue, Face Powder, Eye Makeup Remover</i>	50%
<i>Liquid Foundation, Face Powder, Eye Makeup Remover</i>	25%

7. L3 (large 3-Itemset) (*Eyeliner Blue, Face Powder, Eye Makeup Remover*).
8. Berhenti pada tahap ini karena sudah tidak ada lagi kandidat untuk 4-*Itemset*. Untuk mencari aturan asosiasi diperlukan juga nilai *minimum*

confidence. Misal *minimum confidence* : 75%, aturan asosiasi yang mungkin terbentuk dapat ditunjukkan pada tabel 2.7.

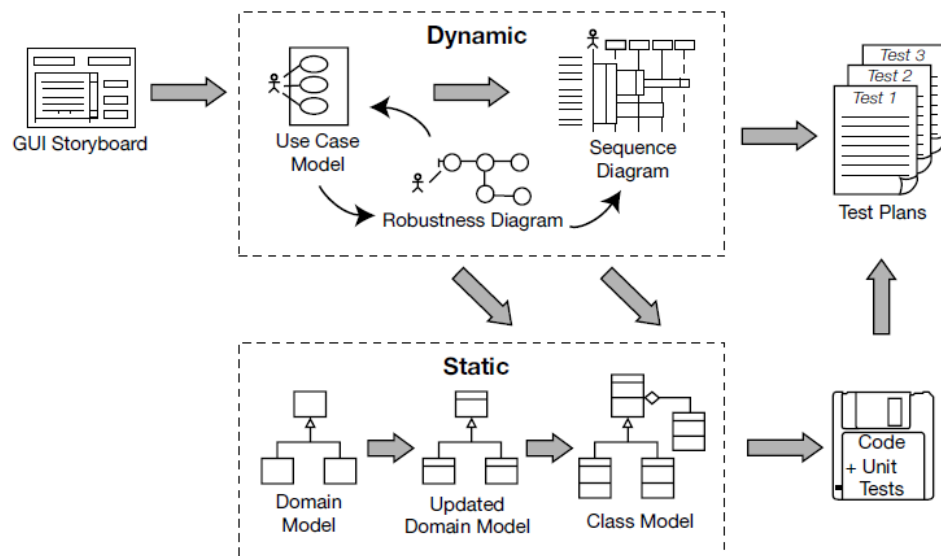
Pada tabel 2.7 telah diketahui data yang telah memenuhi nilai *minimum confidence*. Dari keterangan tersebut dapat diketahui bahwa konsumen yang membeli *Eyeliners Blue* dan *Face Powder* bersamaan dengan *Eye Makeup Remover* memiliki nilai *support* sebanyak 50% dan nilai *confidence* 100%. Dan konsumen yang membeli *Face Powder* dan *Eye Makeup Remover* bersamaan dengan *Eyeliners Blue* memiliki nilai *support* 50% dan nilai *confidence* 100%. Dan konsumen yang membeli *Liquid Foundation* bersamaan dengan *Face Powder* memiliki nilai *support* 50% dan nilai *confidence* 100%. Dan untuk konsumen yang membeli *Eyeliners Blue* bersamaan dengan *Eye Makeup Remover* memiliki nilai *support* 75% dan nilai *confidence* 100%. Begitu juga untuk konsumen yang membeli *Eye Makeup Remover* bersamaan dengan *Eyeliners Blue* memiliki nilai *support* 75% dan nilai *confidence* 100%.

Tabel 2. 9 Hasil Data Yang Memenuhi Nilai *Minimum Confidence*

Aturan (X→Y)	Sup(X ∪ Y)	Sup (X)	Conf	Convic
<i>Eyeliners Blue, Face Powder</i> → <i>Eye Makeup Remover</i>	50%	50%	100%	∞
<i>Face Powder, Eye Makeup Remover</i> → <i>Eyeliners Blue</i>	50%	50%	100%	∞
<i>Liquid Foundation</i> → <i>Face Powder</i>	50%	50%	100%	∞
<i>Eyeliners Blue</i> → <i>Eye Makeup Remover</i>	75%	75%	100%	∞
<i>Makeup Remover</i> → <i>Eyeliners Blue</i>	75%	75%	100%	∞

2.6 ICONIX

Pada Gambar 1 terlihat langkah-langkah proses pemodelan UML dengan menggunakan metode *Use case Driven Object (Iconix process)*.



Gambar 2. 8 Bagan Proses Pemodelan *ICONIX* [17]

ICONIX membagi proses menjadi 2 aliran kerja yaitu *Static* dan *Dynamic* yang saling berhubungan terus-menerus dan beriterasi. Bagian yang static berhubungan dengan struktur sedangkan dynamic berhubungan dengan behaviour. Langkah-langkah yang dilakukan adalah sebagai berikut [17]:

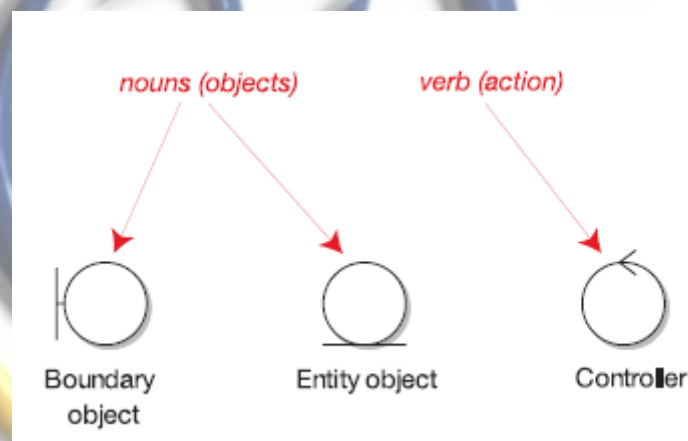
1. *Requirement*

- a. *Functional Requirement*. Mendefinisikan hal-hal yang dilakukan oleh sistem. Kebutuhan ini dapat didefinisikan oleh konsumen/klien bersama dengan analisis
- b. *Domain modeling*. Pemodelan awal untuk membangun istilah yang dipakai dalam proyek. Tujuannya adalah membuat semua orang mempunyai pandangan yang sama tentang permasalahan dengan istilah yang sama. Domain model juga menunjukkan ruang lingkup dan sebagai dasar pembuatan *use case*.
- c. *Behavioral requirement*. Mendefinisikan bagaimana sistem akan berinteraksi. Yang dilakukan adalah desain GUI Storyboard dan identifikasi *use case* yang muncul (*use case modeling* - bagaimana user berinteraksi dengan sistem dan bagaimana sistem merespon).
- d. *Milestone 1: Requirement Review*. Proses kontrol terhadap kesesuaian *use case* dengan kebutuhan klien.

2. Analisis / Preliminary Design (dilakukan pada tiap *use case*)

a. *Robustness analysis*. *Robustness diagram* digunakan sebagai jembatan yang menghubungkan antara proses analisa dan desain. Pada *robustness diagram*, dituliskan juga *use case* deskripsinya dan digambarkan.

Robustness diagram adalah semacam hibrid antara diagram kelas dan diagram aktivitas. Ini adalah representasi bergambar dari perilaku yang dijelaskan oleh *use case*, menunjukkan kelas yang berpartisipasi dan perilaku perangkat lunak, meskipun sengaja menghindari menunjukkan kelas mana yang bertanggung jawab atas bit perilaku mana. Setiap kelas diwakili oleh ikon stereotip grafis. Namun, *robustness diagram* lebih mirip dengan *activity diagram* atau *flowchart*, dalam arti bahwa satu objek "berbicara" dengan objek berikutnya. Alur aksi ini diwakili oleh garis antara dua objek yang saling berbicara.



Gambar 2. 9 *Robustness Diagram Symbols*[17]

Tiga stereotip kelas yang ditunjukkan pada **Gambar 2.9** adalah sebagai berikut [17]:

- *Boundary object* : "antarmuka" antara sistem dan dunia luar. *Boundary object* biasanya adalah layar atau halaman web (misalnya : Presentasi layer yang berinteraksi dengan aktor).
- *Entity object* : Kelas dari model domain.
- *Controller* : Pengikat/ yang menjembatani antara *boundary object* dan *entity object*.

Sangat berguna untuk menganggap *boundary object* dan *entity object* sebagai kata benda, dan pengontrol sebagai kata kerja. Untuk aturannya, berikut saat menggambar *robustness diagram*[17] :

- Kata benda dapat berbicara dengan kata kerja (dan sebaliknya).
 - Kata benda tidak dapat berbicara dengan kata benda lain.
 - Kata kerja dapat berbicara dengan kata kerja lain.
- b. *Update domain model*. Saat membuat *use case* dan *robustness diagram*, jika ada obyek baru yang muncul , dapat dituliskan pada domain model.
 - c. Memberi nama semua fungsi-fungsi logic (controller pada domain model).
 - d. Menuliskan ulang draft awal *use case*.
3. *Milestone 2: Preliminary Design Review*
4. Detailed Design
- a. *Sequence Diagram*. Sequence diagram akan menggambarkan bagaimana *use case* bekerja secara detail dan kronologis. Fungsi utama sequence diagram adalah mengalokasikan behavior dari *use case*.
 - b. *Update domain model*. Saat menggambarkan sequence diagram, update domain model dapat dilakukan dengan menambahkan operation (*method, function, message*) pada domain object. Pada tahap ini, domain object menjadi domain *class* atau entity dan domain model menjadi static model atau *class diagram*.
 - c. Bersihkan static model.
5. *Milestone 3: Critical design Review*
6. *Implementation*
- a. *Coding dan testing*.
 - b. *Integration and scenario testing*.
 - c. *Code review and model update*.

2.7 Unified Modeling Language (UML)




Unified modeling language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal yang membantu pendeskripsian dan desain 15

sistem perangkat lunak khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corps. UML menyediakan notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan [18].

2.6.1 Use case Diagram

Use case Diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case diagram* lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* merepresentasikan sebuah interaksi antara aktor dengan sistem.



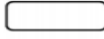


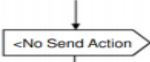



Tabel 2. 10 Notasi *Use case Diagram*

Notasi	Keterangan	Simbol
<i>Actor</i>	Pengguna sistem atau yang berinteraksi langsung dengan sistem, misalnya manusia, aplikasi atau objek lainnya	
<i>Use case</i>	Aktivitas atau sarana yang disediakan oleh bisnis/system	 use case
<i>Association</i>	Sebuah garis yang berfungsi menghubungkan <i>actor</i> dengan <i>use case</i>	

2.6.2 Activity Diagram

Menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Tabel 2. 11 Notasi Activity Diagram





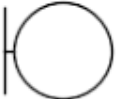
Notasi	Keterangan	Symbol
Initial State	Titik awal untuk suatu aktivitas	
Final State	Titik akhir untuk mengakhiri aktivitas	
Activity	Menandakan sebuah aktivitas	
Decision	Pilihan untuk mengambil keputusan	
Fork / Join	Menunjukkan kegiatan menggabungkan dua panel activity menjadi satu atau satu panel activity menjadi dua	
Send	Tanda pengiriman	
Receive	Tanda penerimaan	
Note	Catatan khusus untuk sebuah aktivitas	
Control Flow	Arus aktivitas	

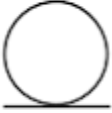

2.6.3 Class Diagram

Class Diagram adalah model statis yang mendukung pandangan statis dari sistem yang sedang berkembang. Ini menunjukkan kelas dan hubungan di antara kelas-kelas yang tetap konstan dalam sistem dari waktu ke waktu. *Class Diagram* sangat mirip dengan ERD, namun *Class Diagram* menggambarkan kelas, yang meliputi atribut, perilaku, dan status, sementara entitas dalam ERD hanya menyertakan atribut [19].

Tabel 2. 12 Notasi Class Diagram

Simbol	Nama	Keterangan
<div style="border: 1px solid black; padding: 5px;"> <p>Nama Class</p> <p>+ atribut</p> <p>+ atribut</p> <p>+ atribut</p> <hr/> <p>+ method</p> <p>+ method</p> </div>	Class	Class adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari class. Bagian tengah mendefinisikan property/atribut class. Bagian akhir mendefinisikan methodmethod dari sebuah class.

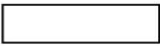
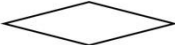


	<i>Association</i>	<p>Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i>. Garis ini bisa melambangkan tipe-tipe relationship dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah relationship. (Contoh: One-to-one, one-to-many, many-to-many).</p>
	<i>Composition</i>	<p>Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi Composition terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah relationship composition digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.</p>
	<i>Dependency</i>	<p>Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut dependency. Umumnya penggunaan dependency digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah dependency dilambangkan sebagai sebuah panah bertitik-titik.</p>
	<i>Aggregation</i>	<p>Aggregation mengindikasikan keseluruhan bagian relationship dan biasanya disebut sebagai relasi</p>
 <p>Boundary</p>	<i>Boundary object</i>	<p>"Antarmuka" antara sistem dan dunia luar. <i>Boundary object</i> biasanya adalah layar atau halaman web (misalnya:</p>

		Presentasi layer yang berinteraksi dengan aktor).
 Entity	<i>Entity object</i>	Kelas dari model domain.
 Control	<i>Controller</i>	Pengikat/yang menjembatani antara <i>boundary object</i> dan <i>entity object</i> .

2.6.4 *Entity Relationship Diagram*

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang termasuk dalam aktifitas pengembangan sistem. ERD adalah suatu model data yang dikembangkan berdasarkan objek.

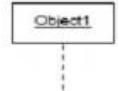




Tabel 2. 13 Notasi *Entity Relationship Diagram*

Simbol	Keterangan
	Entitas, yaitu kumpulan dari objek yang dapat diidentifikasi secara unik
	Relasi, yaitu hubungan yang terjadi antara salah satu lebih entitas. Jenis hubungan antara lain. one to one, One to many, dan many to many.
	Atribut, yaitu karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas.
	Hubungan antara entitas dengan atributnya dan himpunan entitas dengan himpunan relasinya.

2.6.5 *Sequence Diagram*

Menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Tabel 2. 14 Notasi *Sequence Diagram*

Notasi	Keterangan	Symbol
Object	Instance dari sebuah class yang dituliskan tersusun secara horizontal diikuti lifeline	
Activation	Indikasi dari sebuah objek yang melakukan suatu aksi	
Lifeline	Indikasi keberadaan sebuah objek dalam basis waktu	
Message	Indikasi untuk komunikasi antar objek	
Self - Message	Komunikasi kembali ke dalam itu sendiri	

2.8 *My Structured Query Language (MySQL)*

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management sistem*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. MySQL adalah Relational *Database Management Sistem (RDBMS)* yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial [20].

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database (DBMS)* dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh *single user*, kecepatan

query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase*. MySQL juga memiliki beberapa keistimewaan yang lain, antara lain :

1. *Portabilitas*. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan lain-lain.
- b. *Open Source*. MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
- c. *Multiuser*. MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- d. *Performance tuning*. MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
- e. Jenis Kolom. MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
- f. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
- g. Keamanan. MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.

Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

2.9 Review Penelitian Terdahulu

Review penelitian terdahulu dapat dilihat pada **Tabel 2.13** sebagai berikut :

Tabel 2. 15 Review Penelitian Terdahulu

NO.	JUDUL	METODE	HASIL	PERBANDINGAN
1.	ANALISA DATA MINING MARKET BASKET ANALYSIS UNTUK MENGETAHUI POLA BELI KONSUMEN [4]	Metode <i>Data mining</i> dengan teknik Algoritma Apriori	Laporan pembuktian hubungan antar <i>item</i> barang menggunakan Tanagra 1.4	Teknik algoritma Apriori di rancang bangun menggunakan Tanagra 1.4
2.	MARKET BASKET ANALYSIS PADA MINI MARKET AYU DENGAN ALGORITMA APRIORI [3]	Metode <i>Data mining</i> dengan teknik Algoritma Apriori	Laporan pembentukan kombinasi 2 <i>itemset</i> menggunakan Tanagra 1.4	Teknik algoritma Apriori di rancang bangun menggunakan Tanagra 1.4
3.	IMPLEMENTATION OF APRIORI AIGORITHM FOR DETERMINING PURCHASE PATTERNS IN ONE TRANSACTION [21]	Metode <i>Data mining</i> dengan teknik Algoritma Apriori	Laporan Pengujian algoritma Apriori	Teknik algoritma Apriori
4.	IMPLEMENTASI ALGORITMA APRIORI UNTUK MARKET BASKET ANALYSIS BERBASIS R [22]	Metode <i>Data mining</i> dengan teknik Algoritma Apriori	Pembuktian algoritma Apriori untuk <i>Market basket analysis</i> menggunakan R	Teknik algoritma Apriori di rancang bangun menggunakan R
5.	IMPLEMENTASI DATA MINING PADA PENJUALAN EKSPOR LAKBAN DENGAN ALGORITMA APRIORI DI PT. TERAOKA SEISAKUSHO INDONESIA [23]	Metode <i>Data mining</i> dengan teknik Algoritma Apriori	Laporan mendapatkan data-data untuk proses pembentukan <i>generate prequent itemset</i> dan <i>generate association rule</i> dengan parameter berupa periode, <i>threshold</i> , <i>minimum support</i> , dan <i>minimum confidence</i>	Teknik algoritma Apriori di rancang bangun menggunakan MS Visual Studio 2008

6.	<i>APLIKASI MARKET BASKET ANALYSIS DENGAN METODE ASSOCIATION RULE UNTUK MENEMUKAN PERILAKU KONSUMEN MELALUI DATA TRANSAKSI</i> [24]	Metode <i>Data mining</i> dengan teknik Algoritma Apriori	Laporan yang dihasilkan penerapan <i>aplikasi market basket analysis</i> dengan metode <i>association rule</i> menggunakan algoritma Apriori berjalan dengan baik	Teknik algoritma Apriori di rancang bangun menggunakan Netbeans IDE 7.4
----	---	---	---	---

